AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

Claim 1 (Currently amended): A method for optimizing ~~a representation of~~ a code sequence, comprising:

scanning the code sequence to determine a static frequency of operations performed in the code sequence;

determining an executed frequency of operations within multiple times executed loops for the code sequence;

providing a representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations, the representation of operation frequency being a frequency distribution;[[and]]

tuning an instruction set for assigning an op-code representation to an instruction,

wherein the tuning of the instruction set is based on the representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations; and

wherein the optimization of the code sequence may be executed by a microcode loadable from an external source.

Claim 2 (Original): The method of claim 1, wherein the representation of a code sequence is a bit symbol representation.

Claim 3 (Original): The method of claim 1, wherein the instruction set is a variable length instruction set.

Claim 4 (Original): The method of claim 1, wherein the instruction set is a constant length instruction set.

Claims 5-8 (Canceled)

Claim 9 (Previously presented): The method of claim 1, wherein the frequency distribution is a histogram.

Claim 10 (Original): The method of claim 1, wherein a more compact version of the code sequence is accomplished through shortening of bit symbol representation of an op-code of the instruction set.

Claim 11 (Currently amended) A method for optimizing ~~the representation of~~ a code sequence, comprising:

determining [[the]]a frequency of operations performed in the code sequence;

providing a representation of operation frequency, which represents the frequency of operations, the representation of operation frequency being a frequency distribution;

providing a plurality of pre-determined instruction sets with each pre-determined instruction set comprising instructions including assigned op-code representations, each pre-determined instruction set being optimized for a frequency of a particular operation; [[and]]

wherein the frequency of the particular operation includes an executed frequency of operations within multiple times executed loops for the code sequence;

selecting one of the plurality of ~~predetermined~~ pre-determined instruction sets based on the determined frequency of operations performed in the source code sequence; and

wherein the optimizing the code sequence may be executed by a microcode loadable from an external source.

Claim 12 (Original): The method of claim 11, wherein the representation of a code sequence is a bit symbol representation.

Claim 13 (Original): The method of claim 11, wherein the instruction set is a variable length instruction set.

Claim 14 (Original): The method of claim 11, wherein the instruction set is a constant length instruction set.

Claims 15-18 (Canceled)

Claim 19 (Previously presented): The method of claim 11, wherein the frequency distribution is a histogram.

Claim 20 (Original): The method of claim 11, wherein a more compact version of the code sequence is accomplished through shortening of bit symbol representation of an op-code of the instruction set.

Claim 21 (Currently amended): A method for optimizing ~~the representation of~~ a code sequence, comprising:

determining a frequency of use of a register based on scanning the code sequence to determine a static frequency of operations in the code sequence;

modifying the static frequency of operations by analyzing multiple time executed loops to determine an executed frequency of operations in the register for said code sequence;

providing a representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations, the representation of operation frequency being a frequency distribution; [[and]]

tuning an instruction set for assigning a target-code representation for the register,

wherein the tuning of the instruction set is based on the frequency of use of the register,

wherein the tuning of the instruction set is based on the representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations for said register; and wherein the optimizing the code sequence may be executed by a microcode loadable from an external source.

Claim 22 (Original): The method of claim 21, wherein the representation of a code sequence is a bit symbol representation.

Claim 23 (Original): The method of claim 21, wherein the instruction set is a variable length instruction set.

Claim 24 (Original): The method of claim 21, wherein the instruction set is a constant length instruction set.

Claims 25-28 (Canceled)

Claim 29 (Previously presented): The method of claim 21, wherein the frequency distribution is a histogram.

Claim 30 (Currently amended): A method for optimizing ~~the representation of~~ a code sequence, comprising:

determining [[the]] frequency of use of one or more registers within a plurality registers by the operations performed in the code sequence;

providing a representation of operation frequency, which represents the frequency of operations, the representation of operation frequency being a frequency distribution;

wherein the frequency of operations includes an executed frequency of operations within multiple times executed loops for the code sequence;

determining number of registers needed to execute operations; [[and]]

tuning the instruction set for assigning a target-code representation for one or more of the plurality of registers,

wherein the tuning [[of]] the instruction set is based on the frequency of use of the plurality of registers and the number of registers needed to executed operations; and

wherein the optimizing the code sequence may be executed by a microcode loadable from an external source.

Claim 31 (Original): The method of claim 30, wherein the representation of a code sequence is a bit symbol representation.

Claim 32 (Original): The method of claim 30, wherein the instruction set is a variable length instruction set.

Claim 33 (Original): The method of claim 30, wherein the instruction set is a constant length instruction set.

Claims 34-37 (Canceled)

Claim 38 (Previously presented): The method of claim 30, wherein the frequency distribution is a histogram.

Claim 39 (Currently amended): A computer readable medium upon which is stored computer readable code for optimizing a code sequence, wherein said computer readable code upon being executed on a computer causes steps comprising:

scanning the code sequence to determine a static frequency of operations in the code sequence;

determining an executed frequency of operations within multiple times executed loops for the code sequence;

providing a representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations, the representation of operation frequency being a frequency distribution; and

tuning an instruction set for assigning an op-code representation to an instruction,

wherein the tuning of the instruction set is based on the representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations; and

wherein the optimizing of the code sequence may be executed by a microcode loadable from an external source.

Claim 40 (Original): The computer readable medium of claim 39, wherein the representation of a code sequence is a bit symbol representation.

Claim 41 (Original): The computer readable medium of claim 39, wherein the instruction set is a variable length instruction set.

Claim 42 (Original): The computer readable medium of claim 39, wherein the instruction set is a constant length instruction set.

Claims 43-56 (Canceled)

Claim 47 (Previously presented): The computer readable medium of claim 39, wherein the frequency distribution is a histogram.

Claim 48 (Currently amended): An optimized computing assembly, comprising:

a processor coupled with a memory for executing programs; and

an optimized code generator operationally coupled with the processor and the memory, the optimized code generator for scanning a code sequence to determine a static frequency of operations in the code sequence, for determining an executed frequency of operations within multiple times executed loops for the code sequence, for providing a representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations, the representation of operation frequency being a frequency distribution; and for tuning an instruction set by assigning an op-code representation selected according to the executed frequency of operations to an instruction,

wherein the tuning of the instruction set is based on the representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations, and

wherein the optimization of the code sequence may be executed by a microcode loadable from an external source.

Claim 49 (Original): The optimized computing assembly of claim 48, wherein the representation of a code sequence is a bit symbol representation.

Claim 50 (Original): The optimized computing assembly of claim 48, wherein the instruction set is a variable length instruction set.

Claim 51 (Original): The optimized computing assembly of claim 48, wherein the instruction set is a constant length instruction set.

Claims 52-55 (Canceled)

Claim 56 (Previously presented): The optimized computing assembly of claim 48, wherein the frequency distribution is a histogram.

Claim 57 (Currently amended): An optimized code generator for generating a source code stored on a computer readable medium and configured to be executed by a computer, comprising:

a read executable for reading the source code;

a translation executable operationally coupled with the read executable, the translation executable for translating the source code to an intermediate code;

a scanning executable operationally coupled with the translation executable, the scanning executable for determining a static frequency of operations and an executed frequency of operations by analyzing loops performed by the source code and providing a representation based on the static frequency of operations and the executed frequency of operations, the representation of operation frequency being a frequency distribution;

an optimizing translation executable operationally coupled with the scanning executable, the optimizing translation executable for translating the intermediate code to an object code including an optimized instruction set based on the static frequency of operations and the executed frequency of operations;

wherein the optimizing translation may be executed by a microcode loadable from an external source; and

a write executable operationally coupled with the optimizing translation executable, the write executable for outputting an optimized object code.

Claim 58 (Original): The optimized code generator of claim 57, comprising a compiler.

Claim 59 (Original): The optimized code generator of claim 57, comprising an assembler.

Claim 60 (Original): The optimized code generator of claim 57, wherein the instruction set is a variable length instruction set.

Claim 61 (Original): The optimized code generator of claim 57, wherein the instruction set is a constant length instruction set.

Claim 62 (Canceled)

Claim 63 (Previously presented): The optimized code generator of claim 57, wherein the frequency distribution is a histogram.

Claim 64 (Currently amended): An optimized code generator for generating a source code stored on a computer readable medium and configured to be executed by a computer, comprising:

means for scanning the code sequence to determine a static frequency of operations in a code sequence;

means for determining an executed frequency of operations within multiple times executed loops for the code sequence;

means for providing a representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations, the representation of operation frequency being a frequency distribution; and

means for tuning an instruction set by selecting an op-code representation from a plurality of pre-determined sets to an instruction,

wherein the tuning of the instruction set is based on the representation of operation frequency, which represents the static frequency of operations and the executed frequency of operations,

wherein the generating the source code may be executed by a microcode loadable from an external source.

Claim 65 (Original): The optimized code generator of claim 64, comprising a compiler.

Claim 66 (Original): The optimized code generator of claim 64, comprising an assembler.

Claim 67 (Original): The optimized code generator of claim 64, wherein the instruction set is a variable length instruction set.

Claim 68 (Original): The optimized code generator of claim 64, wherein the instruction set is a constant length instruction set.

Claim 69 (Canceled)

Claim 70 (Previously presented): The optimized code generator of claim 64, wherein the frequency distribution is a histogram.